2

(19) The Japanese Patent Office
(12)Laid-Open Patent Application Publication (A)
(11)Laid-Open Patent Application Publication No.Hei4-251297
(43)Publication Date: September 7, 1992

Int. Cl.[5]	Classification Symbol	JPO Ref. No.	FI
G10H 7/10	L	8946-5H	
G10L 9/02		8622-5H	G10 7/00

Request for Examination: not yet requested
Number of Claims: 7 (11 pages in total)16

(21)Application No. Hei3-162514
(22)Application Date: June 7, 1991
(72)Inventor: Tetsuya NIAHIMOTO
c/o Yamaha Corp., No. 10-1, Nakazawa-cho,
Hamamatsu-shi, Shizuoka-ken, Japan
(31)Priority No.: Hei2-410847
(32)Priority Date: December 15, 1990
(33)Priority Country: Japan (JP)
(71)Applicant: Yamaha Corp.
No. 10-1, Nakazawa-cho, Hamamatsu-shi,
Shizuoka-ken, Japan
(74)Agent: Patent Attorney, Yoshihito IIZUKA

(54)Title of the Invention: MUSICAL TONE SYNTHESIZER

[00015]

[Embodiment]

Referring to accompanying drawings, an embodiment of the present invention is described in detail. FIG. 1 is a block diagram, showing a hardware constitution as an embodiment of the electronic musical instrument according to the present invention. In this embodiment, the musical tone synthesizer as a whole is controlled with a microcomputer. The microcomputer includes a microprocessor unit (CPU) 10; a program ROM 11 for storing a system program and various parameters or the like that are not required to be changed; and

a data and working RAM 12 that temporarily stores various data and is used as a working RAM.

[0016]

Preset values of formant sequence are stored in the program ROM 11. Regions for storing the order of the formant sequence, pitch sequence, and voiced and voiceless sound level sequence are assigned to the data and working RAM 12. Thus, the preset values of the formant sequence in the program ROM 11 are loaded by the initialization process starting when power is turned on, so that the musical tone synthesizer starts synthesizing musical tones according to the formant sequence from the time power is turned on.

[0017]

To this microcomputer are connected, through a data and address bus 13, various devices such as a keyboard circuit 14, an operation panel 15, a voiced formant synthesizing sound source 17V and a voiceless formant synthesizing sound source 17U as sound sources, a sequence memory 20, an after-touch sensor 21, and a timer 22. These devices are respectively controlled with the microcomputer.

[0018]

The keyboard circuit 14 includes a circuit made of a plural number of key switches provided so as to correspond to respective keys which specify pitches of musical tones to be produced. Sound producing process for assigning a depressed key to any of a plural number of sound producing channels according to the output from the keyboard circuit 14 is performed with the microcomputer. A process of producing an initial touch data ITD by determining the key depression speed is also performed as required when a key is depressed. An after-touch sensor 21 for outputting after-touch data ATD related to each key of the keyboard by detecting the key depression force when a key is depressed and held is also provided next to the keyboard circuit 14.

[0019]

The operation panel 15 includes various operation members

for choosing, setting and controlling tone color, volume, pitch, and effect. The operation panel 15 has a tone color choosing section 16 for choosing tone colors corresponding to various natural musical instruments such as the piano, organ, violin, brass, guitar, etc. and various other tone colors (human voices). The tone color choosing section 16 outputs tone color choosing signals TC.

[0020]

The voiced formant synthesizing sound source 17V is capable of: simultaneously producing musical tone signals in a plural number (n pieces) of channels; receiving various data given through the data and address bus 13 for keys assigned to respective channels, the data including key code KC data, formant parameter data (center frequency data F and its level data L), relative pitch data P, voiced sound level data V (or level ratio UV of voiced to voiceless sound levels), and other data (such as key-on KON, key-off KOF, initial touch data ITD, after-touch data ATD, and tone color choosing signal TC); and producing musical tone signals according to such data.

[0021]

In other words, the formant synthesizing sound source 17V receives: the center frequency data F, level data L, relative pitch data P, and voiced sound level data V (level ratio UV), of each formant, which have been read from the sequence memory 20; and the key code KC given, as the data specifying the pitch of the musical tone to be synthesized, from the keyboard circuit 14, performs formant synthesizing operation, and outputs a musical tone signal characterized by the formant at a level corresponding to the voiced sound level data V (level ratio UV) corresponding to the pitch specified with the key code KC and the relative pitch data P. The pitch of the musical tone is obtained by multiplying the pitch corresponding to the key code KC by the value of the relative pitch data P. As a matter of course, the pitch may be obtained by other method than multiplication. This embodiment will be described assuming that the number of channels in which the voiced formant

synthesizing sound source 17V can produce sounds simultaneously is 8.

[0022]

A sound source 17U for voiceless formant synthesizing is capable of: simultaneously producing musical tone signals in a plural number (n pieces) of channels; receiving formant parameter data (center frequency data F and its level data L), voiceless sound level data U (or 1 - UV), and other data (tone color choosing signal TC, etc.) given through the data and address bus 13; and producing musical tone signals according to such data.

[0023]

In other words, the formant synthesizing sound source 17U receives the center frequency data F, level data L, voiceless sound level data U (or 1 - UV), of each formant read from the sequence memory 20; performs specified formant synthesizing operation, and outputs a musical tone signal characterized by the formant at a level commensurate with the voiceless sound level data U (or 1 - UV). This embodiment is described on the premise that the number of channels in which the sound source 17U for voiceless formant synthesizing can produce sounds simultaneously is 8, the same as that with the sound source 17V for voiced formant synthesizing.

[0024]

Digital musical signals produced with both of the sound sources 17V and 17U for voiced formant synthesizing are converted to analog musical sound signals through a digital-to-analog (D-A) converters 18V and 18U, and outputted through a sound system 19. The sound system 19 is made up of a speaker, an amplifier, etc. to produce musical sounds commensurate with the analog musical sound signals coming from the D-A converters 18V and 18U.

[0025]

The timer 22 is to give interrupt signals regularly to the microcomputer. With this embodiment, the processes of reading formant parameter data, relative pitch data, and voiced

and voiceless sound level data, to be described later, are implemented with the timer interrupt. The sequence memory 20 has in store various formant data such as the center frequency data F and level data L for a plural number of formants, relative pitch data, voiced and voiceless sound level data, in the sequence corresponding to intended formant change mode.

[0026]

FIG. 2 shows the storage state (example of memory structure) in the sequence memory 20 of the formant parameter data (center frequency data F and level data L), relative pitch data P and voiced and voiceless sound level data V and U. In FIG. 2, the vertical axis corresponds for example to phonemes of human voice identified with sequence numbers X (from 1 to m). The horizontal axis corresponds to time-serial change in: the center frequency and level of the formant characterizing the phoneme identified with the sequence number X of the vertical axis, pitch, voiced sound level, and voiceless sound level, stored in the order of step numbers Y (from 1 to n).

[0027]

In one step of the sequence memory 20 are stored four pieces of formant parameter data FLXYN (with X denoting a sequence number, Y a step number, and N a number indicating any one of four in one step) consisting of center frequency data F and level data L characteristic of the phoneme of the sequence number X. Also stored are, corresponding to the formant parameter data FLXYN, one for each: relative pitch data PXY representing the relative amount of change in the pitch to a reference pitch (identified with a key code KC) at the time the formant is pronounced; voiced sound level data VXY representing the voiced sound level at the time the formant is pronounced; voiceless sound level data UXY (X denoting a sequence number, and Y a step number) representing the level of a voiceless sound. In this case, the value of n of the step number Y in respective sequence numbers X need not necessarily be the same but may be different from one phoneme to another.

[0028]

The sequence memory 20 may store voiced to voiceless sound level ratio data UV in place of the voiced sound level data VXY and the voiceless sound level data UXY. This voiced to voiceless sound level ratio data UV represents the ratio of the voiced sound to the sum of the voiced level and the voiceless level assumed to be 1. Therefore, the voiceless sound level may be easily obtained by subtracting this level UV from 1, as 1 - UV. Storing the voiced to voiceless sound level ratio data UV in the sequence memory 20 is advantageous in that the amount of data in the sequence memory 20 is greatly reduced.

[0029]

Data read from the step number 1 of the sequence number 1 of the sequence memory 20 are: four pieces of formant parameter data FL111, FL112, FL113, and FL114 respectively made up of center frequency data F and level data L; relative pitch data P11, voiced sound level data V11, and voiceless sound level data U11. Data read out from the step number n are: four pieces of formant data FL1n1, FL1n2, FL1n3, and FL1n4; a piece of relative pitch data P1n, voiced sound level data V1n, and voiceless sound level data U1n. These data are read out in the order of the step number Y.

[0030]

Formant parameter data stored in one step may be adapted to be read either in parallel or in time division. Formant parameter data FL111, FL121, ..., FL1n1 read in the order of steps are formant sequence data that change in time series, with their center frequencies and levels changing subtly in time series.

[0031]

Relative pitch data P11, P12, ..., P1n read in the order of steps are pitch sequence data that change in time series and that represent the amounts of change in pitch from the reference pitch. Voiced sound level data V11, V12, ..., V1n and voiceless sound level data U11, U12, ..., U1n, read out in the order of steps are voiced and voiceless sound level sequence data that change in time series. Formant parameter data, relative pitch

data, and voiced and voiceless sound level data of respective sequence numbers are constituted similarly.

[0032]

Reading out of the sequence memory 20 is controlled with a sequence number address data X and a step address data Y. The sequence number address data X is to specify the sequence number of a series of sequence formant data and the sequence pitch data to be read from the sequence memory 20. The step address data Y is to specify a step to be currently read out in the sequence specified with the sequence number address data X. As an example, the sequence number address data X is produced in response to an appropriate choosing operation at the operation panel 15, for example in linked motion with the tone color choosing section 16 or in response to a dedicated sequence number choosing operation means. The step address data Y is produced with the CPU 10 and inputted to an address in the sequence memory 20. Producing this step address data will be described later.

[0033]

This embodiment is described as an example on the premise that the formant parameter data are two in number, i.e. the center frequency data F and the level data L, and that one musical tone signal is synthesized from four formants using the voiced formant synthesizing sound source 17V and the voiceless formant synthesizing sound source 17U.

[0034]

Formant parameter data stored in the sequence memory 20 are produced using voice waveform analysis method of conventional description. For example, formant data extracted with a method such as linear prediction analysis (LPC), line spectrum pair analysis (LSP), and complex sine wave model analysis (CSM) are used.

[0035]

Relative pitch data are also extracted at the same time when the formant data are extracted. Extraction of the relative pitch is performed by modified auto correlation or zero

cross-correlation, for example. In other words, when an original voice is to be sampled, a subject is requested to produce a voice of a pitch that is close to a predetermined reference pitch. This voice contains fluctuation component of the pitch. Then, the difference between the reference pitch and the subject pitch is extracted as a relative pitch data. Extracting voiced and voiceless levels is also made similarly to extracting the relative pitch data. Extracting the level ratio UV may be made using Fourier analysis to obtain a ratio of frequency higher than 5 kHz to that below it.

[0036]

The operation panel 15 has a mode switching key (not shown) for choosing one out of a plural number of reading patterns. The CPU 10 produces step address data that change with time to read out formant parameter data, relative pitch data, and voiced and voiceless level data for a plural number of steps according to a reading pattern chosen with the mode switching key. Producing the step address data according to this reading pattern is made according to musical tone producing timing in response to the key-on signal KON from the keyboard circuit 14. The step address data may either be stored in advance in the data and working RAM 12 according to the reading pattern or produced by numerical operation.

[0037]

FIG. 3 is a schematic view, showing one example for values of voiced sound level data V, voiceless sound level data U, and voiced to voiceless sound level ratio UV, stored in the sequence memory 20. FIG. 3(a) shows the voiced sound level data V, FIG. 3(b) the voiceless sound level data U, and FIG. 3(c) the voiced to voiceless sound level ratio UV. In FIGs. 3(a) to 3(c), the horizontal axis represents the step address of the sequence memory 20, and the vertical axis the level values of the voiced and voiceless sound corresponding to the step address.

[0038]

The levels of voiced sound and voiceless sound levels in FIG. 3 respectively correspond to the cases in which human

voices [a], [ka], [sa], and [ta] are synthesized with a musical tone synthesizer. In FIG. 3(a), all the waveforms represent the level Va of a voiced sound (vowel) [a]. In FIG. 3(b), the first waveform represents the level Uk of a voiceless sound (consonant) [k]; the second, the level Us of a voiceless sound (consonant) [s]; and the third, the level Ut of a voiceless sound (consonant) [t], respectively. While the magnitude of the voiced sound level Va in FIG. 3(a) is constant, the magnitudes of the voiceless sound levels Uk, Us, and Ut in FIG. 3(b) are respectively different. Therefore, the musical tone synthesizer can produce human voices [a], [ka], [sa], and [ta] by reading out the waveforms of FIGs. 3(a) and 3(b) in succession in the order of the step address.

[0039]

FIG. 3(c) shows the voiced to voiceless sound level ratio UV stored in the sequence memory 20. In this example, the ratio of the voiced sound level to the sum of the voiced sound level and the voiceless sound level, assumed to be 1, is stored as the voiced to voiceless sound level ratio in the sequence memory 20. In other words, the voiced to voiceless sound level ratio UV is V/(U + V), where V is the voiced sound level and U is the voiceless sound level. Therefore, the magnitude of the voiceless sound level may be easily calculated as (1 - UV) (indicated with broken line in the figure). However, in case the sum of the voiced sound level and the voiceless sound level is not 1 as shown in FIGs. 3(a) and 3(b), as a matter of course it is preferable to store the voiced sound level V and the voiceless sound level U separately in the sequence memory 20. It is also a matter of course even in case the sum of the voiced sound level and the voiceless sound level is not 1, the ratio between them (V/U) may be simply stored.

[0040]

FIG. 4 shows several example patterns of reading the sequence memory 20. In the reading pattern shown in FIG. 4(a), the step address is increased, in succession, from a specified reference address to a specified maximum address (MAX) and held

there. For example, on the assumption that the reference address is the step number 1 and the maximum address is the step number n, the step address increases in the order of the step numbers 1, 2, 3, . . . , n with time, and held at n once it is reached. In case for example the sequence number is 1, formant parameter data FL11N, FL12N, . . . , FL1nN; relative pitch data P11, P12, . . . , P1n; voiced sound level data V11, V12, . . . , V1n; and voiceless sound level data U11, U12, . . . ,U1n, which numbers correspond to the step numbers respectively, are read out in succession, and finally FL1nN, P1n, V1n, and U1n are read out continuously. Incidentally, the term address in the following description is supposed to denote the term step address.

[0041]

According to the reading pattern shown in FIG. 4(b), reading starts from the reference address to a specified loop end address (LOOP END) in specified sequence, thereafter repeating increase in address starting from the loop start address (LOOP START) toward the loop end address (LOOP END).

[0042]

According to the reading pattern shown in FIG. 4(c), reading starts from the reference address to a specified loop end address (LOOP END) in specified sequence, followed by repetition of decrease in address starting from the loop end address (LOOP END) toward the loop start address (LOOP START).

[0043]

According to the reading pattern shown in FIG. 4(d), reading starts from the reference address to a specified loop end address (LOOP END) in specified sequence, followed by repetition of increase and decrease in address between the loop end address (LOOP END) and the loop start address (LOOP START).

[0044]

The reading patterns in FIG. 4 are examples. It is possible to combine these patterns arbitrarily or to variably control the reading speed according to key operation speed information such as the magnitude of the initial touch data ITD and the after-touch data ATD. It is also possible to remarkably

enhance performance effect by pre-registering various patters according to kinds of musical instruments and kinds of human voices.

[0045]

It is also possible to provide a manual operation member on the operation panel 15 to set any step address, or sequentially change the address by changing the amount of manual operation. As such operation members, there are for example a modulation wheel for outputting signals in 128 steps, and a pitch bend wheel for outputting signals having positive and negative directionality.

[0046]

Next, an example process implemented with the microcomputer is described in reference to FIGs. 5, 6, 7, 8, and 9. FIG. 5 shows details of every step of "main process routine" implemented with the microcomputer. This main process routine is implemented sequentially in the following steps.

[0047]

Step 31: All the data of the microcomputer at the time of turning on power are set to specified values. For example, initial values such as the sequence number address data, step address data, and reading pattern are set to respective registers. The preset values of the formant sequence stored in the program ROM 11 as described before are loaded to the data and working RAM 12.

[0048]

Step 32: Key switches in the keyboard circuit 14 are scanned.

Step 33: Presence or absence of a key event is determined from the results of the key scanning in the step 32. A key depression is determined to be a key-on event, and a key release is determined to be a key-off event. When a key event is present (YES), the process goes to the next step 34; when not present (NO), it goes to the step 35.

[0049]

Step 34: Sound production assigning process is implemented according to the kind of the key event in step 33.

Step 35: Operation members of the operation panel 15 are scanned.

Step 36: Presence or absence of panel events with the operation member is determined from the result of panel scanning in the step 35. For example, whether or not a mode switching key has been pressed is determined. In case a panel event is present (YES), the process goes to the next step 37, or in case a panel event is absent (NO), it returns to the step 32.

Step 37: Sound production assigning process is implemented according to the result of the panel event in the step 36.

[0050]

FIG. 6 shows details of steps in the "sound production process" implemented with the microcomputer. This sound production process is implemented in the following order of steps.

Step 41: Whether the key event is in key-on state or key-off state is determined. In case the event is in key-on state (YES), the process goes to the next step 42; in case it is in key-off state (NO), the process goes to the step 47.

Step 42: A vacant channel is searched in which neither the voiced formant synthesizing sound source 17V nor the voiceless formant synthesizing sound source 17U currently implement a sound production assigning process.

[0051]

Step 43: Presence or absence of vacant channel is determined according to the result of the vacant channel search in the step 42. In case it is present (YES), the process jumps to step 45. In case absent (NO), the process goes to the next step 44.

Step 44: When no vacant channel is present, a truncation process is implemented to form a channel that can implement a sound production assigning process.

[0052]

Step 45: Speed of step address reading phase is determined according to the initial touch data ITD, and is set to the speed register SP (ch). This speed register SP (ch) is provided in each channel, with different value set to each channel. As the reading speeds of the formant sequence, pitch sequence, voiced and voiceless level sequence are different by every key depression, an effect is provided that sounds can be separated even when harmonic keys are depressed.

[0053]

Step 46: A key-on flag is set to the vacant channel so that sound production information in interrupt process is controlled, and the process returns. In other words, no sound production assigning process request is made directly in this sound production process.

Step 47: In case the key event in the step 41 results in key-off, a determination is made if the voiced formant synthesizing sound source 17V and the voiceless formant synthesizing sound source 17U have implemented sound production process of the corresponding key, namely if a corresponding key-on is present. If present (YES), the process goes to the step 48; and if not present, the process returns. This is because of the possibility that a depression-processed key is completely halting sound production due to the truncation process in the step 44.

[0054]

Step 48: Key-off information KOF is sent to a channel corresponding to the key-off event.

Step 49: Key-on flag is reset in the channel to which the key-off information KOF was sent.

[0055]

FIG. 7 shows details of steps in the "panel process" implemented with a microcomputer. This panel process is implemented in the following order of steps. Only parts directly related to this invention are shown here.

Step 51: A determination is made if the mode switching key-on the operation panel 15 has been depressed. If depressed (YES),

the process goes to the next step 52; and if not depressed (NO), to the step 56.

[0056]

Step 52: The value of the mode register MOD is increased by 1 (one) and the process goes to the next step 53. In other words, the value of the mode register MODE is increased by 1 (one) every time this mode changing switch is pressed, to choose any of the reading patterns shown in FIG. 4. As a matter of course, the mode changing switch key is not limited to this one but may be any other one that can change its value at will with a slidable or turnable key.

[0057]

Step 53: As the number of reading patterns in FIG. 4 is four, a determination is made if the value of the mode register MODE is "5." If "5," the process goes to the next step 54; otherwise the process jumps to the step 55. It has only to change the value of this step according to the number of the reading patterns.

[0058]

Step 54: When the value of the mode register MODE in the step 53 is a maximum of 5, "1" is stored here in the mode register MODE. Thus, the value of the mode register MODE circulates with the values of 1, 2, 3, and 4 without overflowing as the mode changing switch is depressed.

[0059]

Step 55: Interruption process vector is overwritten according to the value stored in the mode register MODE. This embodiment is adapted to create the waveforms for reading the formant sequence data, pitch sequence data, and voiced and voiceless sound level sequence data by software process and adapted to form respective reading waveforms by different interrupt processes. Therefore, destination of interruption is changed according to the stored value in the mode register MODE.

[0060]

Step 56: When the mode switching key (not shown) on the

operation panel 15 has not been pressed, process is implemented with other keys on the operation panel 15. For example, in case a tone color choosing event occurs, sequence address number change or the like is made according to the tone color.

[0061]

FIG. 8 shows details of the process of reading the formant parameter data, relative pitch data, and voiced and voiceless sound level data implemented with the microcomputer. FIG. 8 shows an interrupt process with the reading pattern shown in FIG. 4(a), or when the stored value of the mode register MODE shown in FIG. 7 is "1." This routine is implemented in the following order of steps every time an interrupt signal is given from the timer 22.

[0062]

Step 61: Interrupt is prohibited to prevent multiple coincident interrupts.

Step 62: Value "1" is set to the channel number register channel.

Step 63: A determination is made if a key-on flag of a channel corresponding to the value stored in the channel number register channel is set. If set (YES), the process goes to the next step 64; and if not set (NO), the process jumps to the step 613 to implement the process of the next channel with the value of the channel number register channel increased by "1."

[0063]

Step 64: The value of the step address register Y (ch) is increased by "1," and further a value SP (ch) x SE, obtained by multiplying the speed register SP (ch) by sensitivity SE, is added. When this sensitivity SE is made "0," the step number Y increases by "1" at a time; when the sensitivity SE is made a positive value, the rate of increase in the step number Y increases, and when the sensitivity SE is made a negative value, the rate of increase in the step number Y decreases.

[0064]

In other words, when the sensitivity SE is "0," formant parameter data and relative pitch data are read from the

sequence memory 20 in the order of step numbers such as FL11N, FL12N, FL13N, ... When the sensitivity SE is a positive value, reading from the sequence memory 20 is made in skipping order of step numbers such as FL11N, FL13N, FL15N, ..., resulting in higher reading speeds of the formant parameter data, relative pitch data, and/or voiced and voiceless sound level data. When negative in contrast, the formant parameter data, relative pitch data, and/or voiced and voiceless sound level data are read for the same step numbers such as FL11N, FL11N, FL12N, FL12N, FL13N, FL13N, ..., resulting in lower reading speed.

[0065]

Incidentally, it may also be adapted to use only the speed register SP (ch) in place of SP (ch) x SE so that positive and negative values commensurate with the touch data may be stored in the speed register SP (ch). It may be alternatively adapted to variably control the values of the sensitivity SE and/or speed register SP (ch) according to the values of the initial touch data ITD and/or after-touch data ATD.

[0066]

Step 65: A determination is made if the sequence number of the step address register Y (ch) is greater than the maximum step value MAX. If greater, the process goes to the step 66; and if smaller, the process goes to the step 67.

Step 66: When no step number greater than the maximum step value MAX of the sequence number is present, the maximum step number MAX of the sequence number is stored in the step address register Y (ch).

[0067]

Step 67: A value 1 is stored in the register N to sequentially read four formant parameter data of each step number Y.

Step 68: Voiced sound level data VXY and voiceless sound level data UXY corresponding to the sequence number X and the step number Y are read from the sequence memory 20 and respectively outputted to the voiced formant synthesizing sound source 17V and the voiceless formant synthesizing sound source

17U. This makes it possible to express subtle changes in the levels of voiced sound and voiceless sound. Further, in case the level ratio UV of the voiced sound of FIG. 3(c) is stored in the sequence memory 20, the level ratio UV is outputted to the voiced formant synthesizing sound source 17V. From the level ratio UV, the voiceless sound level ratio (1 - UV) is calculated and then outputted to the voiceless formant synthesizing sound source 17U.

[0068]

Step 69: Relative pitch data PXY corresponding to the sequence number X and the step number Y are read from the sequence memory 20 and then outputted together with the key code KC to the voiced formant synthesizing sound source 17V. This makes it possible to express subtle displacement in pitch. Incidentally, relative pitch data PXY and the key code KC are not taken into the voiceless formant synthesizing sound source 17U.

Step 610: Formant parameter data FLXY1 corresponding to the sequence number X and the step number Y are read from the sequence memory 20 and then outputted to the voiced formant synthesizing sound source 17V and the voiceless formant synthesizing sound source 17U.

Step 611: The value of the register N is increased by "1."

[0069]

Step 612: If the value of the register N is greater than "4" or not is determined. If greater, as the four formant parameter data FLXY1 - FLXY4 of the step number Y have been read, the process goes to the next step 613. If smaller, as the four formant parameter data have not been read yet, the process returns to the step 610 to implement the process until the four formant parameter data are read.

[0070]

Step 613: The value of the channel number register channel is increased by "1."

Step 614: If the value of the channel number register channel is greater than "8" or not is determined. If greater,

the process goes to the next step 615. If smaller, the process returns to the step 63 to apply the same process to the next channel.

Step 615: The interrupt prohibited in the step 61 is allowed to return to the normal process.

[0071]

FIG. 9, like FIG. 8, shows details of the process of reading formant parameter data, relative pitch data, and voiced and voiceless sound level data implemented with the microcomputer. It shows an interrupt process for the case in which the stored value of the mode register MODE shown in FIG. 7 is "2," or for the case of the reading pattern shown in FIG. 4(b). The steps 71 - 74, 77 - 715 are the same as the steps 61 - 64, 67 - 615. Therefore, their explanations are omitted.

[0072]

Step 75: A determination is made if the value of the step address register Y (ch) is greater or not than the loop end address value LE at the sequence number. If greater, the process goes to the step 76. If smaller, the process goes to the step 77.

Step 76: A loop start address value LS at the sequence number is stored in the step address register Y (ch).

[0073]

Through these steps 75 and 76, it is possible to read formant parameter data, relative pitch data, and voiced and voiceless sound level data in succession according to the reading pattern as shown in FIG. 4(b) from the sequence memory 20. Incidentally, reading patterns of FIGs. 4(c) and 4(d) may be easily realized by modifying the flow of FIGs. 8 and 9. Therefore, their explanations are omitted here.

[0074]

The embodiment described above makes it possible to produce musical tones by changing formants in time series and to change musical tone pitch and voiced and voiceless sound levels in time series. There is also an effect at that time to freely control the rate of change in the formant, pitch, and

voiced and voiceless sound levels by changing the reading speed.

[0075]

Further, any type of musical tone synthesizing method may be used in the voiced formant synthesizing sound source 17V. For example, an amplitude modulation (AM) as described in JP-B-Sho-59-19352 or a frequency modulation (FM) as described in JP-B-Sho-62-42515 may be used. FIG. 10 shows an example of the voiced formant synthesizing sound source 17V that synthesizes the voiced formant according to the amplitude modulation (AM) using a window function.

[0076]

A phase generator 81 shown in FIG. 10 produces phase data corresponding to the center frequency by successively adding up the center frequency data F for setting the formant center frequency. Therefore, the successive addition speed is low when the formant center frequency value is small, and the successive addition speed is high when the formant center frequency value is great. When the added-up value overflows, the value returns to the initial value and repeats successive addition. When a reset pulse RS of a specified time duration is given, the sum value is reset to zero, and output is held to zero for only the specified time duration. The sum output of the phase generator 81 is supplied as address data through a selector 85 to a logarithmic sine (log sin) function table 86.

[0077]

A phase generator 82 is of an accumulator structure and takes in basic pitch frequency data f0 corresponding to the key code KC, and successively adds up the basic pitch frequency data f0. Also this phase generator 82, when its value overflows, returns to the initial value and repeats adding action. The phase generator 82 is adapted to output an overflow pulse (for example a most significant bit MSB) to a differentiating circuit 83.

[0078]

The differentiation circuit 83 is made of a mono-stable

multi-vibrator to output, when an overflow pulse rises up, a reset pulse signal RS of a specified time duration to the phase generators 81 and 84. In other words, the differentiating circuit 83 detects the time point when the output value of the phase generator 82 becomes zero and outputs a reset pulse signal RS at the time point. Therefore, the phase data of the formant center frequency produced with the phase generator 81 are reset for a specified period of time at a frequency corresponding to the pitch of the musical tone to be produced according to the reset pulse RS. As a result, amplitude modulation is performed with the formant center frequency used as the carrier frequency and with the musical tone pitch frequency used as the modulation frequency.

[0079]

The phase generator 84 is a circuit for adding up phoneme modulation wave phase constant K supplied from a tone color parameter supply circuit (not shown) synchronously with specified clock pulses. The phase generator 84 is adapted to retain the final value of the added-up sum when the sum overflows. Then, it resets its contents when a reset pulse RS is supplied and starts adding up again from the initial value. The added-up result of the phase generator 84 is supplied as address data through the selector 85 to the logarithmic sine (log sin) function table 86. Here, the phoneme modulation wave phase constant K is set so that the adding up speed of the phase generator 84 is far lower in comparison with the adding up speed of the phase generator 81.

[0080]

The selector 85 chooses output data of the phase generator 81 when action choice signal SEL is supplied, chooses output data of the phase generator 84 when no action choice signal SEL is supplied, and supplies them as address data to the sine function table 86.

[0081]

The sine function table 86 is a table storing sine function data in logarithmic expression for one period (or may be 1/2

period or 1/4 period) and is adapted to output sine function values in logarithmic expression commensurate with address data supplied through the selector 85. Therefore, the sine function table 86 outputs sine function values at a rate commensurate with the added up value at the phase generator 81 or 84.

[0082]

A data shifter 87 is a circuit that shifts the output data of the sine function table 86 according to shift amount data S which is a tone color parameter. The shift amount data S is supplied also from the tone color parameter supply circuit (not shown). The data shifter 87 performs shifting action when an action signal SFT is supplied; when the action signal SFT is not supplied, outputs data as they are coming from the sine function table 86. The shift in the data shifter 87 is an action of shifting on more significant side by the value of the shift amount data S.

[0083]

An adder 88, when an action signal ADD1 is supplied, adds up the output data of the data shifter 87 and the output data of a register 89. When no action signal ADD1 is supplied, the data supplied to the adder 88 are outputted as they are from an output terminal. The register 89 is also adapted to store data that have passed intact through the adder 88. In this case, addition with the adder 88 is made with logarithmic data, which means multiplication for antilogarithms.

[0084]

An adder 810, when an action signal ADD2 is supplied to it, adds up the output data of the adder 88 and level-converted level data L. The addition with the adder 810 is made with logarithmic data, which means multiplication for antilogarithms.

[0085]

A logarithmic-linear (log-linear) conversion circuit 811 is a circuit that converts data supplied from the adder 810 in logarithmic expression into antilogarithms. Data outputted from the logarithmic-linear conversion circuit 811 are given

to an accumulator 812. Four formant parameter data F and L for synthesizing a musical tone are given in time division, and phoneme signals corresponding to respective formants are outputted in succession from the logarithmic-linear conversion circuit 811, added up in the accumulator 812, and outputted as a musical tone to a multiplier 813. The multiplier 813 receives voiced sound level data V (or UV), multiplies it by a musical tone signal coming from the accumulator 812, and outputs the result as a musical tone signal of a voiced sound.

[0086]

The above description is on an example of the voiced formant synthesizing sound source 17V. As detailed description on the voiced formant synthesizing sound source 17V appears in the specification of Japanese patent application No. Hei-1-77383, it is omitted here. Additionally, in the voiced formant synthesizing sound source 17V in FIG. 10, the phase generator 84, the selector 85, and the data shifter 87 may be omitted.

[0087]

FIG. 11 shows an example of the voiceless formant synthesizing sound source 17U. As details of this voiceless formant synthesizing sound source 17U appear in the specification of Japanese patent application No. Hei-2-271397, simplified description is given here. In FIG. 11, a white noise generating circuit 91 is to generate white noise having a flat spectrum. A digital filter 92 is a low-pass filter called IIR filter to convert flat spectrum white noise into noise having a specified band width. The digital filter 92 is made up of: an inverter 93, a band width parameter generator 94, a delay circuit 95, adders 96, 97, 98, and a multiplier 99. The digital filter 92 converts white noise into noise signal having a spectrum characteristic sloping down toward the right and outputs it to a multiplier 910.

[0088]

A periodic waveform generating circuit 911 outputs a sinusoidal sequential sample point amplitude value *sin2■fot*

having a formant center frequency *fo* transitioning according to center frequency data F for setting formant center frequency. The periodic waveform generating circuit 911 is made up of a phase accumulator 912 and a sine table 913. The phase accumulator 912 adds up center frequency data F synchronously with specified clock pulses. As the center frequency data F correspond to the formant center frequency *fo* of noise sound intended to be produced, the phase accumulator 912 outputs its added-up value as a reading address signal of the sine table 913.

[0089]

The sine table 913 is a table having stored sine function data for one period (or may be 1/2 or 1/4 period) to be read out with reading address signals. Therefore, from the sine table 913 comes out a sine wave of a frequency *fo* corresponding to the reading address signal (added-up value) of the phase accumulator 912. The multiplier 910 multiplies the noise signal of the digital filter 92 by the sine wave of the periodic waveform generating circuit 911 and outputs the product. Therefore, noise signal having specified formant characteristic comes out of the multiplier 910.

[0090]

An envelope generator 914 outputs envelope signals, for controlling the amplitude of noise signals outputted from the multiplier 910, synchronously with clock pulses and according to the level data L of the formant parameter, to a multiplier 915. The multiplier 915 multiplies the envelope signal by the noise signal from the multiplier 910 and outputs the product to a next-step multiplier 916. The multiplier 916 receives the voiceless sound level data U (or 1 - UV), multiplies it by the noise signal from the multiplier 915, and outputs the product as the musical sound signal of the voiceless sound.

[0091]

While the above embodiment is described as realized with software, embodying the invention is not limited to the above but may be realized with hardware. It may also be adapted that

specific values of formant parameter data in the sequence memory 20 can be arbitrarily overwritten by operation on the operation panel 13.

[0092]

The above embodiment is described on the assumption that the step number of FIG. 2 is read according to the reading pattern of FIG. 4. The manner of reading is not limited to the above. That is, formant parameter data may be read out sequentially according to the reading pattern of FIG. 4, moving from the step number to the sequence number. Additionally, while the above embodiment is described assuming that all of the formant parameter data, relative pitch data, and voiced and voiceless sound level data are read, it may be assumed that any one kind of data are read.

[0093]

Amounts of change in the pitch and the voiced and voiceless sound levels are small in comparison with the formant parameter data, and the speed of change is also low. Therefore, relative pitch data and voiced and voiceless sound levels need not be provided for every step number. Instead, it may be adapted to provide relative pitch data and voiced and voiceless sound levels at intervals of several steps, or to store formant parameter data, relative pitch data, and voiced and voiceless sound levels in different sequence memories to be read at different sequence speeds, or to provide different step numbers.

[0094]

Relative pitch data used may be extracted by analysis, or produced with a dedicated editor or the like. A switch may be provided to bring the output of the relative pitch data or voiced and voiceless sound level data to an off state; or a switch may be provided to vary the depth (magnitude) of the relative pitch data or voiced and voiceless sound level data. This is in preparation for a possible need of producing a flat sound without any change both in pitch and in voiced and voiceless sound levels. As for the formant parameter data, like the pitch

data, it may be adapted that the amount of change relative to a certain reference formant data are stored sequentially in time series.

[0095]

While the above embodiment is described assuming that parameters of the center frequency and level for specifying formant are stored as formant parameter data in the sequence memory 20, in case the formant is synthesized by frequency modulation operation, it may be adapted to have in store various parameters for that purpose such as the center frequency, modulation frequency, modulation index, and level, to be read out. As a matter of course, this applies to synthesizing formants of rhythm sounds or the like as well as scale sounds.

[0096]

[Effects of the Invention]

According to the invention, formant parameter data, relative pitch data, or voiced and voiceless sound level data, changing in time series, are pre-stored over a plural number of steps in a memory means. The formant parameter data, relative pitch data, or voiced and voiceless sound level data are sequentially read with a reading means over a plural number of steps. This makes it possible to naturally change the formant, pitch or voiced and voiceless sound levels of musical tones like actual musical instrument sound or human voice.

[Brief Description of Drawings]

FIG. 1 is a block diagram, showing a hardware constitution as an embodiment of the electronic musical instrument according to the present invention.

FIG. 2 shows the state of the formant parameter data and relative pitch data stored in the sequence memory shown in FIG. 1.

FIG. 3 schematically shows an example of values of voiced sound level data and voiceless sound level data stored in the sequence memory.

FIG. 4 shows several example patterns of reading the sequence memory shown in FIG. 1.

FIG. 5 shows a flowchart of an exemplary main routine implemented with the microcomputer shown in FIG. 1.

FIG. 6 shows a flowchart of detailed sound production process of FIG. 4 implemented with the microcomputer of FIG. 1.

FIG. 7 shows detailed example of panel process of FIG. 4 implemented with the microcomputer of FIG. 1.

FIG. 8 shows a flowchart of detailed process of reading formant parameter data, relative pitch data, and voiced and voiceless sound level data according to the reading pattern of FIG. 3(a).

FIG. 9 shows a flowchart of detailed process of reading formant parameter data, relative pitch data, and voiced and voiceless sound level data according to the reading pattern of FIG. 3(b).

FIG. 10 shows an example of voiced formant synthesizing sound source of FIG. 1.

FIG. 11 shows an example of voiceless formant synthesizing sound source of FIG. 1.

[Description of Reference Numerals and Symbols]

10: CPU
11: program ROM
12: data and working RAM
13: data and address bus
14: keyboard circuit
15: operation panel
16: tone color choosing section
17V: voiced formant synthesizing sound source
17U: voiceless formant synthesizing sound source
18: D-A converter
19: sound system
20: sequence memory
21: after-touch sensor
22: timer

FIG. 1

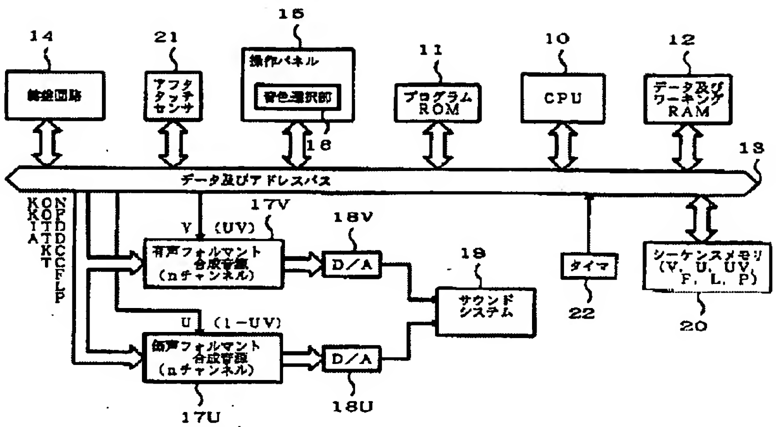


FIG. 5

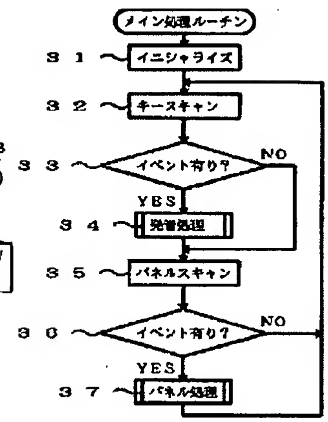


Fig. 1
10: CPU
11: Program ROM
12: Data & working RAM
13: Data & address bus
14: Keyboard circuit
15: Operation panel
16: Tine color choosing section
17U: Voiceless formant synthesizing section (n channels)
17V: Voiced formant synthesizing section (n channels)
18U: D/A
18V: D/A
19: Sound system
20: Sequence memory
21: After-touch sensor
22: Timer

Fig. 5
Main process routine
S1: Initialize
S2: Key scanning
S3: Is event present?

S4: Sound production process
S5: Panel scanning
S6: Is event present?
S7: Panel process

FIG. 3

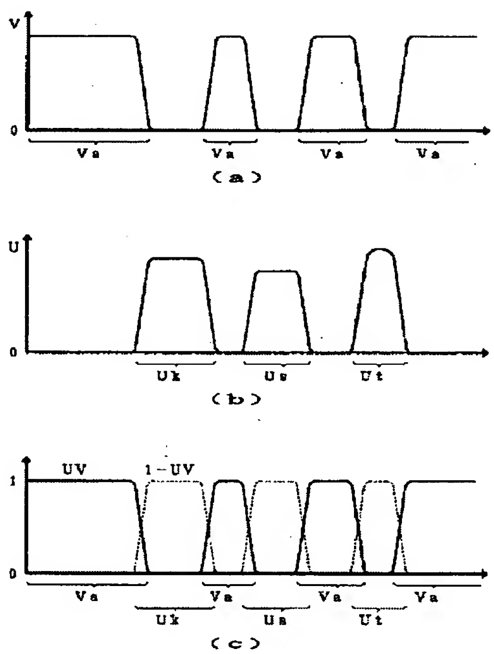


FIG. 6

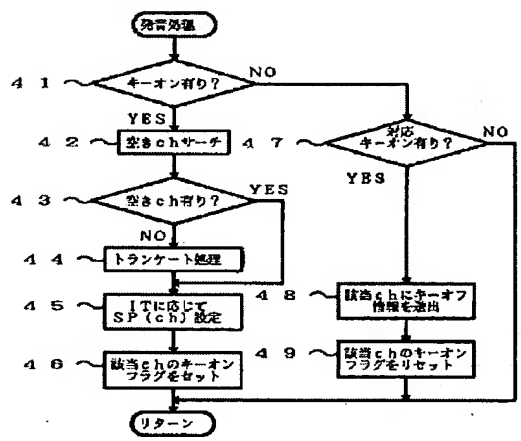
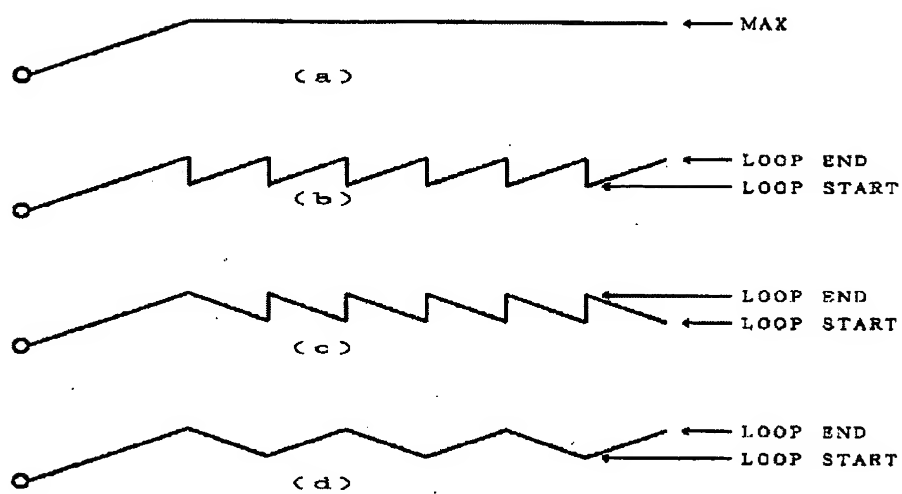


Fig. 6
Sound production process
41: Is key-on present?
42: Vacant channel search
43: Is vacant channel present?
44: Truncate process
45: Set SP(ch) according to IT
46: Set key-on plug of corresponding channel
47: Is corresponding key-on present?
48: Send key-off information to corresponding channel
49: Reset key-on plug of corresponding channel

Return

FIG. 4



MAX
(a)
LOOP END
LOOP START
(b)
LOOP END
LOOP START
(c)
LOOP END
LOOP START
(d)

FIG. 7

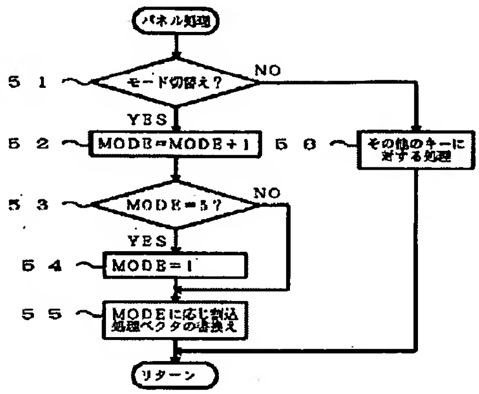


FIG. 8

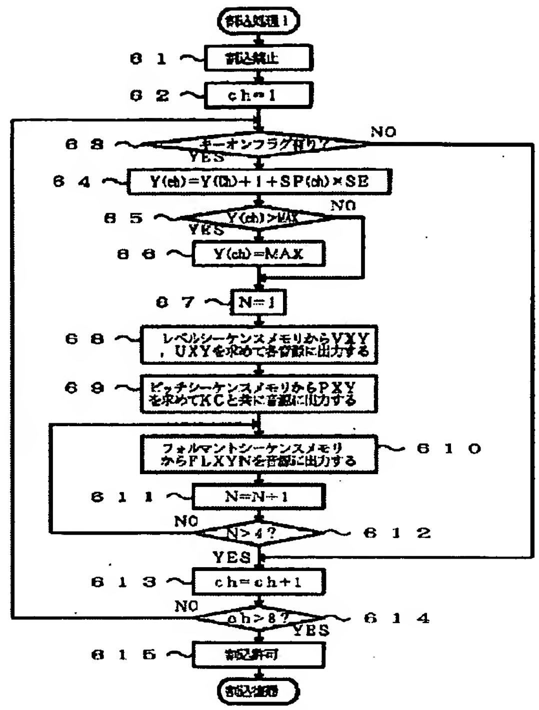


Fig. 7

Panel process

51: Mode switched?

52: MODE=MODE+1

53: MODE=5?

54: MODE=1

55: Rewrite interrupt process vector according to MODE

56: Process to other keys

Return

Fig. 8

Interrupt process 1

61: Interrupt prohibit

62: ch=1

63: Is key-on flag present?

64: Y(ch)=Y(ch)+1+SP(ch)*SE

65: Y(ch)>MAX

66: Y(ch)=MAX

67: N=1

68: Obtain VXY and UXY from level sequence memory, output them to each sound generator

69: Obtain PXY from pitch sequence memory, output it together with KC to sound generator

610: Output FLXYN from formant sequence memory to sound generator

611: N=N+1

612: N>4?

613: ch=ch+1

614: ch>8?

615: Allow interrupt

Return interrupt

FIG. 9

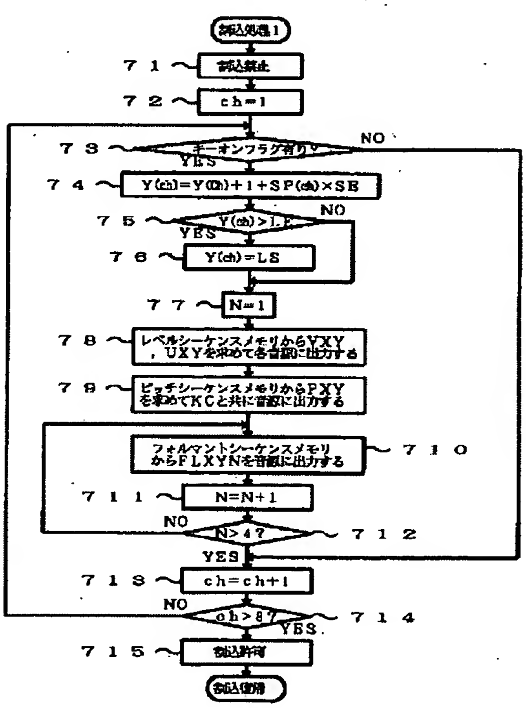


FIG. 10

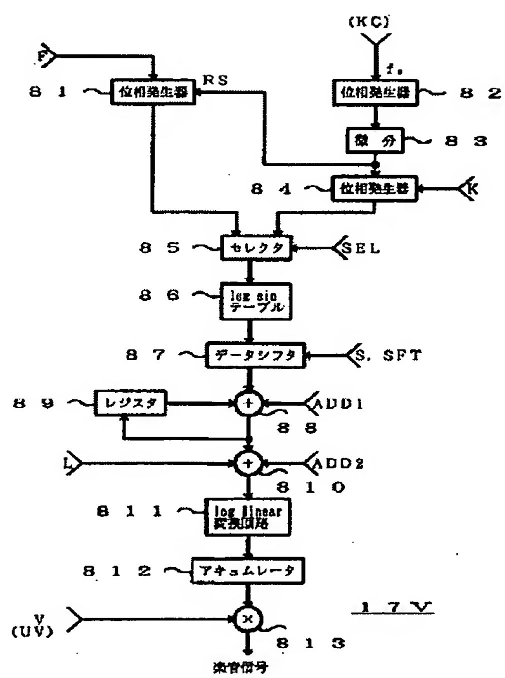


Fig. 9

Interrupt process 1

71: Interrupt prohibit

72: ch=1

73: Is key-on flag present?

74: Y(ch)=Y(ch)+1+SP(ch)*SE

75: Y(ch)>LE

76: Y(ch)=LS

77: N=1

78: Obtain VXY and UXY from level sequence memory, output them to each sound generator

79: Obtain PXY from pitch sequence memory, output it together with KC to sound generator

710: Output FLXYN from formant sequence memory to sound generator

711: N=N+1

712: N>4?

713: ch=ch+1

714: ch>8?

715: Allow interrupt

Return interrupt

Fig. 10

81: Phase generator

82: Phase generator

83: Differentiation

84: Phase generator

85: Selector

86: Log sin table

87: Data shifter

89: Register

811: Log linear conversion circuit

812: Accumulator

Musical tone signal

FIG. 11

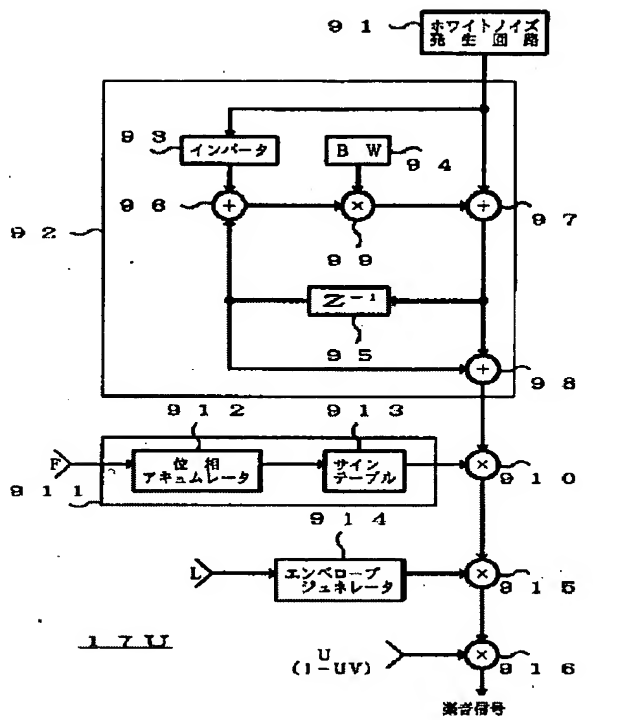


Fig. 11

91: White noise producing circuit

93: Inverter

912: Phase accumulator

912: Sine table

914: Envelope generator

Musical tone signal